APML

Dr. Matan Gavish

Fall 2018

Lecture 1: Manifold Learning (I)

Updated January 2, 2019

האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

Lecture 1: Manifold Learning (I)

1. High Dimensional Data
2. Linear Dimension Reduction
3. Locally linear data

Overview

Goals:

- Understand math foundation of popular data analysis algorithms
- Called “manifold learning” or “nonlinear dimension reduction”
- These methods are used for
  - data visualization
  - data organization
  - clustering
  - preprocessing before standard ML algorithms

    (classification, regression, ranking, etc)

Overview

Who cares?

- These are standard methods in toolbox of any data scientist
- More importantly, they teach a useful **mindset**
- Advice: meditate on the mindset

High Dimensional Data

High-dimensional data

- The data in these lectures is standard arrangement:
- Each data point is $\mathbf{x} \in \mathbb{R}^p$ and we have n of them: $\mathbf{x}_1, \ldots, \mathbf{x}_n$.
- The Euclidean space $\mathbb{R}^p$ is a big place.
- When $p \gg 1$, any direction you take will be orthogonal to any other, almost.
- To know our way around $\mathbb{R}^p$ (think density estimation), we need lots of data.
- There are ways to quantify how much is "lots", but generally, to properly learn a distribution over $\mathbb{R}^p$ you'll need n **exponential** in p.

High-dimensional data

- This is what Bellman called *the curse of dimensionality*.
- Realistically, these days we have $n \sim p$ (or sometimes worse, $p \gg n$.)
- When $n \sim p$ or worse, we say that the problem involves **high-dimensional data**.

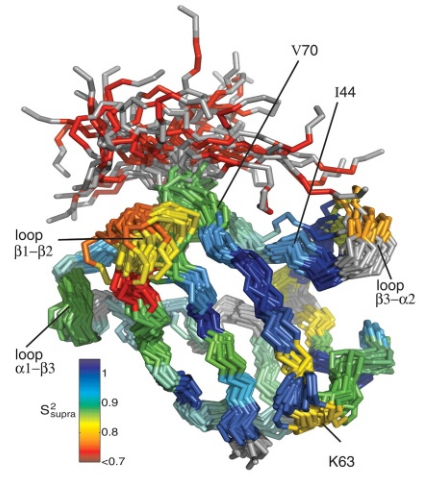
Example: Face tracking

Source: `visagetechnologies.com`

Example: Digit recognition

Source: Liu et al, Handwritten digit recognition, Pattern Recognition 37(2) 2004

Example: Molecular dynamics

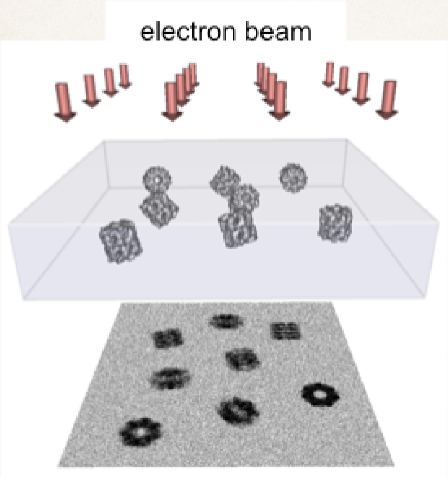


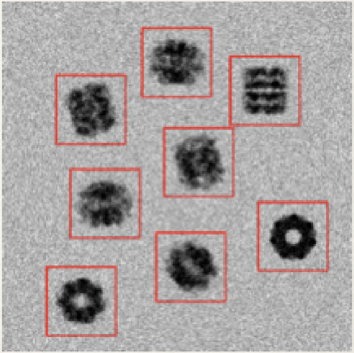
Source: rosettadesigngroup.com/blog/400/molecular-dynamics-vs-monte-carlo/

Cryo-EM microscopy

Source: Reuters

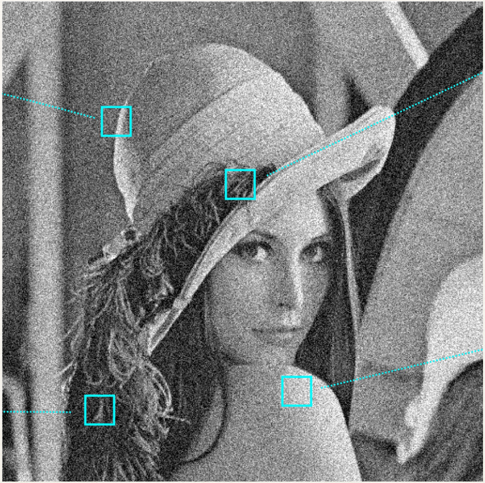
Cryo-EM microscopy





Source: csail.mit.edu

Non-local means image denoising

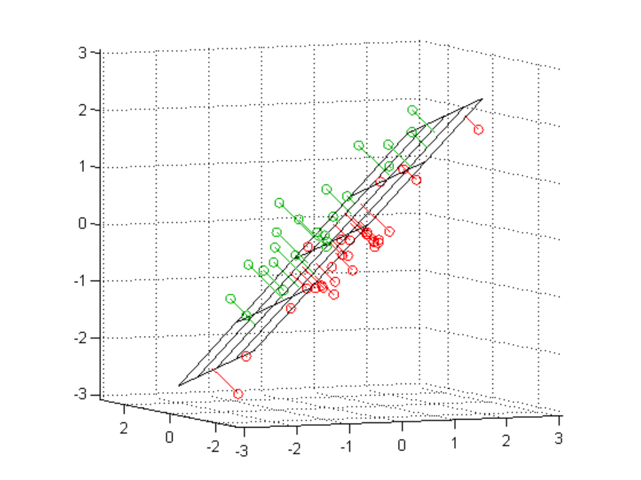


Source: home.deib.polimi.it/boracchi

All is lost?

- How can we analyze data / practice machine learning in high dimensions?
- Often we assume some hidden structure like
  - Sparsity
  - Low rank
  - **Low intrinsic dimension**
- Each deserves an entire course. We will focus on the latter
- To get a first taste, assume first that the data lives on a low-dimensional linear subspace of $\mathbb{R}^p$

Linear Dimension Reduction



Source: winedarksea.org

Setup

- Assume that data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ actually sit on a low-dimensional subspace $V \subset \mathbb{R}^p$, with $dim(V) = d \ll p$.
- So while data may appear to be *p*-dimensional, it isn't
- Sometimes say that *p* is the **ambient** dimension and *d* is the **intrinsic** dimension of the data
- For any of the data-analysis purposes mentioned above, it would be good to **reduce dimentions** from *p* to *d*

Dimension reduction - definition

- What does it mean to *reduce dimensions*
- Want new features that describe the same dataset $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^d$.
- Easier to work with $\mathbf{y}$'s for any task we have in mind
- If $d \ll p$ we escaped the high-dimensional setting
- What are the properties we hope the $\mathbf{y}$'s will have?
- Ideally, $\mathbf{y}_i = f(\mathbf{x}_i)$ for some really good f
- Best possible f: **Isometry**.
- In this case $||\mathbf{y}_i|| = ||\mathbf{x}_i||$ and $||\mathbf{x}_i - \mathbf{x}_j|| = ||\mathbf{y}_i - \mathbf{y}_j||$, $1 \leq i, j \leq n$
- This means that the $\mathbf{y}$'s dataset is equivalent to the $\mathbf{x}$'s dataset for any purpose we might have.

Linear dim-reduction methods

- Principal component analysis (PCA)
- Multidimensional scaling (MDS)

PCA

- In PCA we diagonalize the p-by-p **empirical covariance matrix** of the data

$$S = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$$

- Assume for simplicity that each feature has zero empirical mean so that $\bar{\mathbf{x}} = 0$
- Write $S = U \Lambda U^\top$, where U orthonormal and Λ diagonal
- Let U_d be p-by-d matrix with first d columns of U
- Take $\mathbf{y}_i = \mathbf{x}_i \cdot U_d$
- (Note! $\mathbf{x}_i$ and $\mathbf{y}_i$ are row vectors!)

Homework - PCA

1. Show that the data sits on d dimensional subspace of R^p (namely, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in V \subset \mathbb{R}^p$ **iff** the empirical covariance S is of rank d.
2. Define the intrinsic coordinates in this case via PCA and show how to find them.
3. In this case, how to get an orthonormal basis which spans V by PCA?
4. Show that the $\mathbf{y}$'s defined in the previous slides are the result of an isometry (on the subspace V)

MDS

- Importantly, in PCA we where given the original data $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- In MDS we only want the **distances** $\Delta_{i,j} = ||\mathbf{x}_i - \mathbf{x}_j||^2$.
- **MDS step 1:** Form the n-by-n **similarity matrix**

$$S = -\frac{1}{2} H \cdot \Delta \cdot H \,, \tag{1}$$

where $H = I - \frac{1}{n}\mathbf{1} \cdot \mathbf{1}^\top$ is a data-centering matrix.

MDS

- **MDS step 2:** Diagonalize S to form

$$S = U \cdot \Lambda \cdot U' \tag{2}$$

  where $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$ and U is orthogonal with orthonormal columns $\mathbf{u}_1, \ldots \mathbf{u}_n$.

- **MDS step 3:** Return the n-by-d matrix with columns $\sqrt{\lambda_i}\mathbf{u}_i$ ($i = 1, \ldots, d$). Embed the points into $\mathbb{R}^d$ using the rows of this matrix (namely $\mathbf{y}_i$ is the i-th row)

Homework - MDS

1. Assume that the data sits on d dimensional subspace of R^p (namely, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in V \subset \mathbb{R}^p$. What is the rank of the matrix that MDS diagonalizes?
2. Define the intrinsic coordiantes in this case via MDS and show how to find them.
3. How to get an orthonormal basis which spans V by MDS?
4. Show that the $\mathbf{y}$'s defined in the previous slide (result of MDS) are the result of an isometry (on the subspace V)

Homework - SVD

- Recall the Singular Value Decomposition (SVD)
- Let X be the *n*-by-*p* data matrix whose rows are $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- Let $X = UDV^\top$ be an SVD of X
- Assume again data sits exactly on a *d*-dimensional linear subspace of $\mathbb{R}^p$
- Show how to use *d* first right singular vectors (*d* left columns of *V*) for a linear dimensionality reduction equivalent to the methods using PCA and MDS above
- Convince yourself that this method is basically equivalent to the PCA-based method. What's the difference?

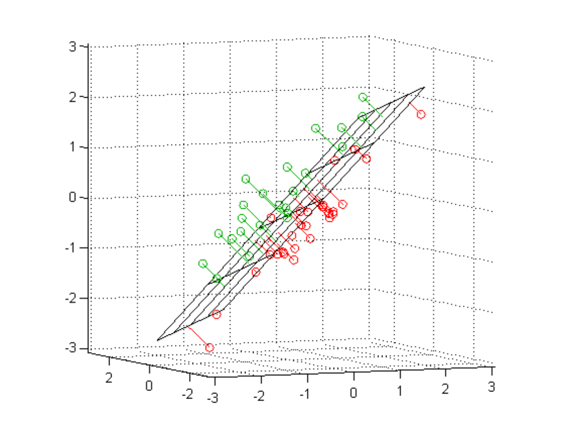
Meditation (I)

Many people don't understand the difference between PCA and MDS. Don't be one of them. Hint: In PCA we get the data points $\mathbf{x}_i$ and diagonalize a p-by-p matrix. In MDS we **only** observe the distances, not the points, and diagonalize an n-by-n matrix.

Meditation (II)

Think long and hard about why **diagonalization** appears in both PCA, MDS and SVD. What's so magical about diagonalization?

Comment



Unfortunately, data are never **exactly** on a subspace.

Source: winedarksea.org

Comment

- PCA and MDS are designed to work well also when the data is **approximately** on a subspace, not exactly on it. This is beyond our present scope.
- Briefly: When data are **approximately** on a *d*-dimensional subspace, using PCA, MDS and SVD with this value *d* will find the subspace and project the data onto it

Wait! How to choose *d* ?

- In practice we never know the subspace dimension
- Prove: If the data sit exactly on a *d* dimensional subspace, then:
  - PCA has exactly *d* non-zero principal values
  - MDS has exactly *d* non-zero eigenvalues
  - SVD has exactly *d* non-zero singular values
- So we can simply infer *d* from the spectrum (eigenvalues)
- The traditional way to visualize the spectrum is called the **Scree Plot** - plot the eigenvalues / principal values / singular values in decreasing order

Let's add ambient noise

- **Ambient noise** is noise that contaminates the data vector in $\mathbb{R}^p$
- The following exercise will help you understand what happens to the Scree Plot when the noise level grows
- This will help us understand how to choose *d* in practice

Homework: The Scree Plot with noise

- Choose specific values for n, p, d. For example you can take $n = 500$,$p = 1000, d = 5$.
- Create n data points $x_1, \ldots, x_n \in \mathbb{R}^p$ that sit exactly on a d-dimensional linear subspace of $\mathbb{R}^p$.
- Here is one way to do this:
  - Draw a n-by-d i.i.d Gaussian matrix
  - Paste a n-by-$p - d$ zero matrix to obtain a n-by-p matrix X. Here the data sit on a d-dimensional linear subspace spanned by the first d vectors of the standard basis in $\mathbb{R}^p$

Homework: The Scree Plot with noise (cont.)

- To rotate the subspace to a random direction, draw a uniformly-at-random rotation matrix in $\mathbb{R}^p$
- This can be done (for example) by running the QR decomposition on an i.i.d Gaussian matrix
- (Educate yourself on the QR decomposition and on the meaning of "uniformly at random rotation" etc
- With X the matrix from the previous slide, let's use XQ where Q is the random rotation matrix

Homework: The Scree Plot with noise (cont.)

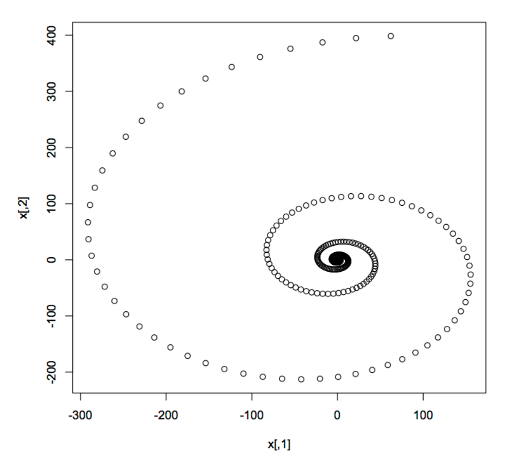
- Now draw a noise matrix Z, say a n-by-p i.i.d Gaussian matrix with mean 0 and variance 1
- Let σ denote a noise level and consider the data matrix $X + \sigma Z$
- Run the three methods for linear dimension reduction (using PCA, MDS, SVD) and plot the scree plot with $\sigma = 0$
- Observe that there are exactly d non-zero eigenvalues in each of the methods
- Now gradually increase σ and see what happens to the Scree Plot
- Observe that the eigenvalues $d + 1$ and onward rise from zero, but when σ is small enough there's a gap
- Observe that when σ is large enough the gap closes

So, how to choose *d*

- Most practitioners look for a “gap” in the scree plot and choose *d* this way
- This is not an algorithm, since we subjectively use our eyes
- Playing with the simulation in the homework above will help you understand this method

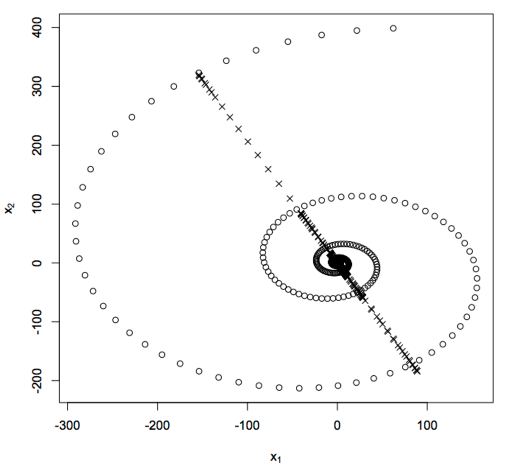
Locally linear data

PCA this data...



Source: Cosma Shalizi 36-350 notes, CMU

Oops.

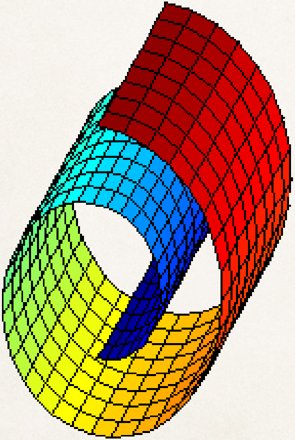


Source: Cosma Shalizi 36-350 notes, CMU

Enter the Swiss Roll

Source: cook.fnr.sndimg.com

Swiss Roll data

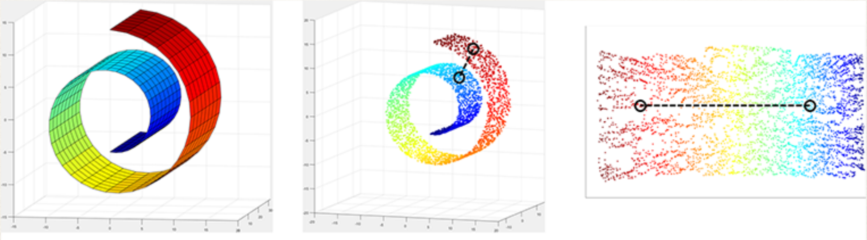


Source:

Euclidean distances: not great

- Should we be able to perform dim-reduction on Swiss Roll data?
- But what will it mean?
- We said that the new coordinates – the **y**'s – are “useful” if the distances $||\mathbf{y}_i - \mathbf{y}_j||$ will be meaningful in some sense to the distances between the **x**'s.
- But which distances between the **x**'s?
- Clearly the **y**'s will **not** be a result of isometry as before.

Euclidean distances: not great



(a) (b) (c)

Source: Wang et al, Head pose estimation, in: Manifolds: current research areas, 2017

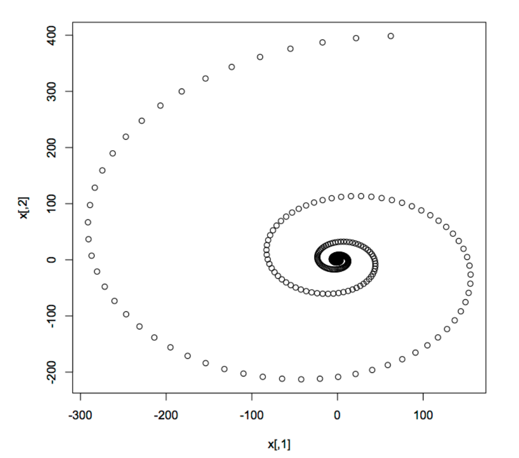
Euclidean distances: not great

- In this example, points close-by in Euclidean distance on original dataset (**x**'s) should **not** be close-by in dim-reduced dataset (**y**'s)
- Also $||\mathbf{x}_i - \mathbf{x}_j||$ doesn't really tell us anything useful (why?)
- More specifically, when the Euclidean distance is large, it is often meaningless - points that are $q \gg 1$ apart are not twice as dissimilar as points that are $2q$ apart.

Manifolds

- A manifold is a fundamental notion in Differential Geometry
- For our purpose, a manifold is a smooth, curved subset embedded in Euclidean space
- (There is another, more abstract way to define a manifold as a topological space without thinking about it as a subset of any Euclidean space.)

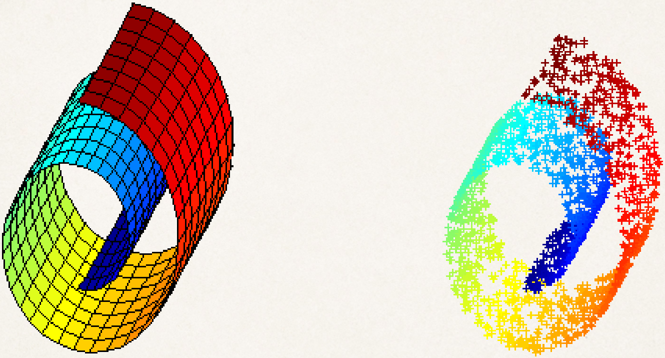
Manifolds



This data is sampled from a 1-dimensional manifold embedded in $\mathbb{R}^2$.

Source: Cosma Shalizi 36-350 notes, CMU

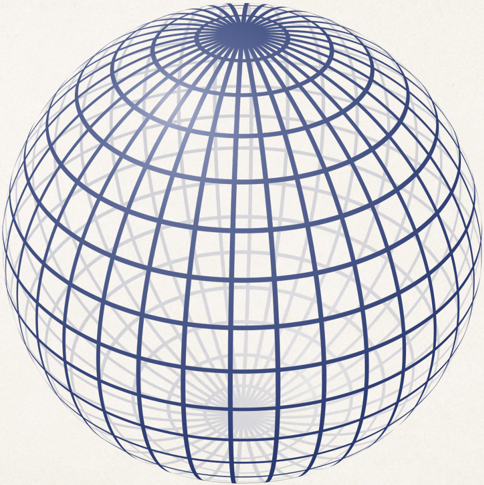
Manifolds



The swiss roll data is sampled from a 2-dimensional manifold embedded in $\mathbb{R}^3$

Source:

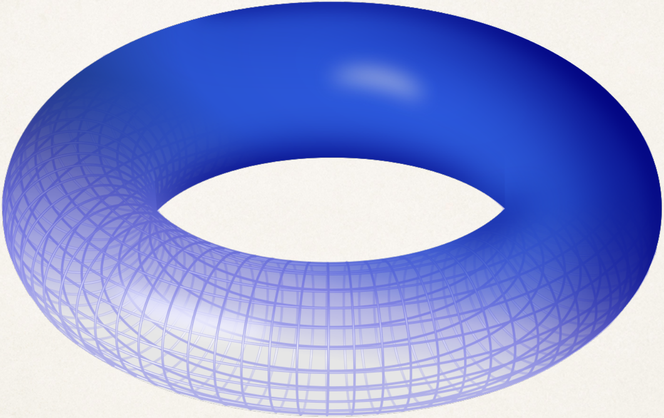
Manifolds



The sphere is a 2-dimensional manifold embedded in $\mathbb{R}^3$

Source: Wikipedia

Manifolds



The torus is a 2-dimensional manifold embedded in $\mathbb{R}^3$

Source: Wikipedia

Manifolds

- The defining property of a q-dimensional manifold is that it can be arbitrarily well-approximated by a q-dimensional linear subspace.
- (Formally, every point has an open neighborhood which is homeomorphic to $\mathbb{R}^q$, and the transition of these homeomorphisms from neighborhood to neighborhood is continuous and differentiable.)
- The notion of angle, surface and volume on a manifold is due to Riemann. He was required to measure areas in a hilly countryside…
- Comment: If you don't want people shutting you up with the words "measure" and "manifold", take measure theory and differential geometry, both beautiful subjects.

Manifold learning

- Unsupervised methods for discovering intrinsic manifold coordinates $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^d$ for data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ that lives on or near a *d*-dimensional manifold.
- What's the connection between **y**'s and **x**'s?
- There's a smooth map *f* (think locally linear *f*) we don't know such that $f(\mathbf{x}_i) = \mathbf{y}_i$
- this means that $||\mathbf{y}_i - \mathbf{y}_j|| \approx ||\mathbf{x}_i - \mathbf{x}_j||$ **if** $||\mathbf{x}_i - \mathbf{x}_j||$ is small.

Meditation (III)

Suppose that there are points on a grid in $\mathbb{R}^p$ and we are interested in a function $f : \mathbb{R}^p \to \mathbb{R}$. We are only given $f(\mathbf{x}_i) - f(\mathbf{x}_j)$ for nearby points $\mathbf{x}_i, \mathbf{x}_j$. Can we recover f? Yes we can - this is called *integration*. In manifold learning we are recovering a global shape from local difference affinity (only use local differences!). The integration machine is *diagonalization*.

Credit

Some content adapted from notes by Amit Singer (Princeton) and Cosma Shalizi (CMU)